Installation Guide for SADC Advanced Sampling Workshop

System Requirements and Environment Setup

Prerequisites

Operating System Requirements

- **Windows**: Windows 10 version 1903 or higher (64-bit)
- **macOS**: macOS 10.13 (High Sierra) or higher
- **Linux**: Ubuntu 20.04 LTS or equivalent distribution

Hardware Requirements

- **Minimum RAM**: 8 GB (16 GB recommended for large datasets)
- **Storage**: 10 GB free space for software and workshop materials
- **Processor**: Multi-core processor (Intel i5 or equivalent minimum)
- **Internet**: Stable connection for package downloads and updates

Step 1: R Installation

Windows Installation

1. Navigate to https://cran.r-project.org/bin/windows/base/
2. Download R-4.3.2 for Windows (64-bit)
3. Run installer with administrative privileges
4. Select installation directory (default recommended: C:\Program Files\R\R-4.3.2)
5. Select all core components during installation
6. Complete installation and verify by opening R console

macOS Installation

1. Navigate to https://cran.r-project.org/bin/macosx/
2. Download R-4.3.2-arm64.pkg for Apple Silicon or R-4.3.2-x86_64.pkg for Intel
3. Open downloaded package and follow installation wizard
4. Verify installation by opening Terminal and typing: `R --version`

Linux Installation (Ubuntu/Debian)

```bash
# Update indices
sudo apt update -qq

# Install helper packages
sudo apt install --no-install-recommends software-properties-common dirmngr

# Add R 4.3 repository
wget -qO- https://cloud.r-project.org/bin/linux/ubuntu/mariadb-repo-key.asc | sudo tee -a /etc/apt/trusted.gpg.d/cran
sudo add-apt-repository "deb https://cloud.r-project.org/bin/linux/ubuntu $(lsb_release -cs)-cran40/"

# Install R
sudo apt install --no-install-recommends r-base r-base-dev

# Verify installation
R --version
```

Step 2: RStudio Installation

All Operating Systems

1. Visit https://posit.co/download/rstudio-desktop/
2. Download RStudio Desktop 2023.09.1 or later for your operating system
3. Run installer with default settings
4. Launch RStudio and verify R integration
5. Configure RStudio settings:
    - Tools > Global Options > General: Uncheck "Restore .RData"
    - Tools > Global Options > Code > Display: Check "Show line numbers"
    - Tools > Global Options > Packages: Set CRAN mirror to nearest location

Step 3: Rtools Installation (Windows Only)

Windows users must install Rtools for compiling packages:

1. Download Rtools43 from https://cran.r-project.org/bin/windows/Rtools/
2. Run rtools43-[version].exe installer
3. Use default installation path: C:\rtools43
4. Add Rtools to PATH during installation

5. Verify in R console:

```r
Sys.which("make")
# Should return: "C:\\rtools43\\usr\\bin\\make.exe"
```

Step 4: Essential System Libraries

Windows

No additional system libraries required (included in Rtools)

macOS

Install Xcode Command Line Tools:

```bash
xcode-select --install
```

Install Homebrew and additional libraries:

```bash
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
brew install gdal proj geos udunits
```

Linux (Ubuntu/Debian)

```bash
```

```
sudo apt-get update
sudo apt-get install -y \
  libcurl4-openssl-dev \
  libssl-dev \
  libxml2-dev \
  libgdal-dev \
  libgeos-dev \
  libproj-dev \
  libudunits2-dev \
  libfontconfig1-dev \
  libharfbuzz-dev \
  libfribidi-dev \
  libfreetype6-dev \
  libpng-dev \
  libtiff5-dev \
  libjpeg-dev
```

Step 5: R Package Installation

Open RStudio and run the following installation script:

```
r
```

```
# Set CRAN repository
options(repos = c(CRAN = "https://cran.rstudio.com/"))

# Install packages in groups to manage dependencies

# Core survey packages
install.packages(c(
  "survey",     # Complex survey analysis
  "sampling",   # Sample selection algorithms
  "sae",        # Small area estimation
  "pps"         # PPS sampling
))

# Data manipulation packages
install.packages(c(
  "tidyverse",  # Complete tidyverse ecosystem
  "data.table", # High-performance data operations
  "janitor",    # Data cleaning utilities
  "haven"       # SPSS/Stata/SAS file import
))

# Visualization packages
install.packages(c(
  "ggplot2",    # Already included in tidyverse
  "plotly",     # Interactive graphics
  "sf",         # Spatial features
  "tmap",       # Thematic maps
  "viridis",    # Color palettes
  "scales"      # Scale functions for graphics
))

# Statistical modeling packages
install.packages(c(
  "lme4",       # Mixed-effects models
  "Matrix",     # Sparse and dense matrices
  "MASS",       # Modern applied statistics
  "boot",       # Bootstrap functions
  "car",        # Regression diagnostics
  "emmeans"     # Estimated marginal means
))

# Reporting and documentation packages
install.packages(c(
  "rmarkdown",   # Dynamic documents
  "knitr",       # Dynamic report generation
  "xaringan",    # Presentation slides
  "kableExtra",  # Table formatting
  "gt",          # Grammar of tables
  "flextable"    # Flexible tables
))

# Additional utility packages
install.packages(c(
  "here",        # Project-relative paths
  "conflicted",  # Conflict resolution
  "devtools",    # Package development tools
  "roxygen2",    # Documentation generation
  "testthat",    # Unit testing
  "bench"        # Performance benchmarking
))

# Verify all installations
required_packages <- c(
  "survey", "sampling", "sae", "pps",
  "tidyverse", "data.table", "janitor", "haven",
  "plotly", "sf", "tmap", "viridis",
  "lme4", "Matrix", "MASS", "boot",
  "rmarkdown", "knitr", "xaringan", "kableExtra",
  "here", "conflicted", "devtools"
)

installed_packages <- rownames(installed.packages())
missing_packages <- setdiff(required_packages, installed_packages)

if(length(missing_packages) == 0) {
  cat("✓ All required packages successfully installed\n")
} else {
  cat("✗ The following packages failed to install:\n")
  print(missing_packages)
}
```

Step 6: Environment Configuration

Create a project-specific .Rprofile file in the workshop directory:

```
r
```

```
# SADC Workshop .Rprofile Configuration

# Set options for reproducibility
options(
  digits = 4,
  scipen = 10,
  width = 80,
  warning.length = 1000,
  stringsAsFactors = FALSE,
  repos = c(CRAN = "https://cran.rstudio.com/")
)

# Load frequently used packages silently
suppressPackageStartupMessages({
  library(tidyverse)
  library(survey)
  library(here)
})

# Set default theme for ggplot2
theme_set(theme_minimal(base_size = 12))

# Custom functions for workshop
source_if_exists <- function(file) {
  if(file.exists(file)) source(file)
}

# Display startup message
cat("════════════════════════════════════════════════════════════\n")
cat("   SADC Advanced Sampling Methods Workshop Environment Loaded   \n")
cat("   R Version:", R.version.string, "\n")
cat("   Working Directory:", getwd(), "\n")
cat("════════════════════════════════════════════════════════════\n\n")

# Set random seed for reproducibility
set.seed(2024)
```

Step 7: System Verification

Run the comprehensive system check script:

```
r
```

```r
# System_Check.R - Complete Environment Verification

# Function to check package version
check_package <- function(pkg, min_version = NULL) {
  if (!requireNamespace(pkg, quietly = TRUE)) {
    return(list(installed = FALSE, version = NA, status = "✗ Not installed"))
  }

  pkg_version <- as.character(packageVersion(pkg))

  if (!is.null(min_version)) {
    if (package_version(pkg_version) < package_version(min_version)) {
      status <- paste("⚠ Outdated (need >=", min_version, ")")
    } else {
      status <- "✓ OK"
    }
  } else {
    status <- "✓ OK"
  }

  return(list(
    installed = TRUE,
    version = pkg_version,
    status = status
  ))
}

# Check R version
cat("R Environment Check\n")
cat("══════════════════════════════════════════\n")
cat("R Version:", R.version.string, "\n")

if (getRversion() < "4.3.0") {
  cat("⚠ Warning: R version 4.3.0 or higher recommended\n")
} else {
  cat("✓ R version meets requirements\n")
}

# Check RStudio (if running in RStudio)
if (Sys.getenv("RSTUDIO") == "1") {
  cat("RStudio Version:", rstudioapi::versionInfo()$version, "\n")
  cat("✓ Running in RStudio\n")
} else {
  cat("ℹ Not running in RStudio\n")
}

cat("\n")

# Check critical packages
cat("Package Installation Status\n")
cat("══════════════════════════════════════════════════\n")

packages_to_check <- list(
  list(name = "survey", min_version = "4.2"),
  list(name = "sampling", min_version = "2.9"),
  list(name = "sae", min_version = "1.3"),
  list(name = "tidyverse", min_version = "2.0.0"),
  list(name = "rmarkdown", min_version = "2.20"),
  list(name = "xaringan", min_version = "0.28"),
  list(name = "sf", min_version = "1.0")
)

results <- data.frame(
  Package = character(),
  Version = character(),
  Status = character(),
  stringsAsFactors = FALSE
)

for (pkg_info in packages_to_check) {
  check <- check_package(pkg_info$name, pkg_info$min_version)
  results <- rbind(results, data.frame(
    Package = pkg_info$name,
    Version = ifelse(is.na(check$version), "—", check$version),
    Status = check$status
  ))
}

print(results, row.names = FALSE)

# Check memory
cat("\nSystem Resources\n")
cat("══════════════════════════════════════════════════\n")
cat("Available Memory:", round(as.numeric(system("wmic OS get TotalVisibleMemorySize /value", intern = TRUE)[2]) / 1
cat("Number of Cores:", parallel::detectCores(), "\n")

# Check working directory and file structure
```

```r
cat("\nProject Structure Check\n")
cat("══════════════════════════════════════════\n")

required_dirs <- c(
  "00-Setup", "01-Data", "02-Scripts", "03-Outputs",
  "04-Presentations", "05-Exercises", "06-Solutions",
  "07-Resources", "08-Harry-Journey"
)

for (dir in required_dirs) {
  if (dir.exists(dir)) {
    cat("✓", dir, "exists\n")
  } else {
    cat("✗", dir, "missing - creating now...\n")
    dir.create(dir, recursive = TRUE)
  }
}

# Test basic survey functionality
cat("\nFunctionality Test\n")
cat("══════════════════════════════════════════\n")

tryCatch({
  # Create simple test data
  test_data <- data.frame(
    id = 1:100,
    stratum = rep(1:5, each = 20),
    weight = runif(100, 0.5, 2),
    y = rnorm(100)
  )

  # Create survey design
  test_design <- survey::svydesign(
    ids = ~1,
    strata = ~stratum,
    weights = ~weight,
    data = test_data
  )

  # Test estimation
  test_mean <- survey::svymean(~y, test_design)

  cat("✓ Survey package functional\n")
  cat("✓ Test estimation completed\n")
}, error = function(e) {
  cat("✗ Error in functionality test:\n")
  cat("  ", e$message, "\n")
})

cat("\n══════════════════════════════════════════\n")
cat("System check complete. Ready for workshop!\n")
cat("══════════════════════════════════════════\n")
```

Step 8: Data Download and Verification

Download workshop datasets from the repository:

```
r
```

```r
# Data_Download.R - Automated data retrieval

# Set data directory
data_dir <- here::here("01-Data")

# List of required data files
data_files <- c(
  "household_survey_main_2024.csv",
  "household_roster_2024.csv",
  "enumeration_areas_master.csv",
  "auxiliary_census_2022.csv",
  "mobile_populations_sample.csv",
  "panel_rotation_cohort_2023.csv",
  "mixed_mode_responses.csv",
  "small_area_indicators.csv"
)

# Base URL for data repository (replace with actual repository)
base_url <- "https://github.com/SADC-Stats/sampling-workshop/raw/main/01-Data/"

# Download each file
for (file in data_files) {
  file_path <- file.path(data_dir, file)

  if (!file.exists(file_path)) {
    cat("Downloading", file, "...\n")
    download.file(
      url = paste0(base_url, file),
      destfile = file_path,
      mode = "wb"
    )
  } else {
    cat("✓", file, "already exists\n")
  }
}

# Verify data integrity
cat("\nVerifying data files...\n")
for (file in data_files) {
  file_path <- file.path(data_dir, file)
  if (file.exists(file_path)) {
    data <- read.csv(file_path, nrows = 5)
    cat("✓", file, "-", nrow(data), "rows preview loaded\n")
  } else {
    cat("✗", file, "not found\n")
  }
}
```

Troubleshooting Guide

Common Installation Issues

Issue 1: Package compilation fails on Windows

- Solution: Ensure Rtools is properly installed and in PATH
- Verify: `Sys.which("make")` should return Rtools path

Issue 2: sf package installation fails

- Solution: Install system libraries (gdal, geos, proj)
- Alternative: Install binary version from CRAN

Issue 3: Permission denied errors

- Solution: Run RStudio as administrator (Windows) or use personal library
- Set personal library: `.libPaths("~/R/library")`

Issue 4: Memory allocation errors

- Solution: Increase memory limit in R
- Windows: `memory.limit(size = 16000)`
- All systems: Use data.table for large datasets

Issue 5: Xaringan slides not rendering

- Solution: Install Chrome or Chromium browser
- Set Chrome path: `pagedown::find_chrome()`

Pre-Workshop Checklist

Before attending the workshop, verify:

□ R version 4.3.0 or higher installed
□ RStudio 2023.06.0 or higher installed
□ All required packages successfully installed
□ System check script runs without errors

□ Workshop materials downloaded and accessible
□ At least 10 GB free disk space available
□ Can create and knit a basic R Markdown document
□ Can read CSV files and create basic plots

Support Contacts

Technical Issues: workshop.tech@sadc-stats.org
Package Problems: Include output of `sessionInfo()` in email
Emergency Contact: +27 XX XXX XXXX (08:00-17:00 SAST)

Additional Resources

- R for Data Science: https://r4ds.hadley.nz/
- Survey Package Documentation: https://r-survey.r-forge.r-project.org/
- RStudio Cheatsheets: https://posit.co/resources/cheatsheets/
- Stack Overflow R Tag: https://stackoverflow.com/questions/tagged/r

---

Document Version: 1.0
Last Updated: [Current Date]
Next Review: [Pre-workshop Date]